

Overview of the PPC64 targets support in Yocto Project




Yocto Project Virtual Summit 2021

Maciej Pijanowski





Maciej Pijanowski
Embedded Firmware Team Leader

-  [@macpijan](https://twitter.com/macpijan)
-  maciej.pijanowski@3mdeb.com
-  [linkedin.com/in/maciej-pijanowski-9868ab120](https://www.linkedin.com/in/maciej-pijanowski-9868ab120)
- 5 years in 3mdeb
- interested in:
 - Embedded Linux
 - build systems (e.g. Yocto)
 - system security



- coreboot licensed service providers since 2016
- coreboot project leadership participants
- UEFI Adopters since 2018
- Official consultants for Linux Foundation fwupd/LVFS project
- Yocto Participants and Embedded Linux experts
- Open Source Firmware enthusiasts and evangelists

- Why ppc64?
- History of the ppc64 support in Yocto
- Overview of the available ppc64 targets
- BE and LE considerations
- Testing qemu_ppc64 (Big Endian)
- Testing qemu_ppc64le (Little Endian)
- Talos II platform overview
- POWER9 boot flow
- Adding support for Talos II hardware target
- Testing build for Talos II
- Q&A

- IBM OpenPOWER architecture
 - open documentation and open source firmware
- 3mdeb is porting coreboot to POWER9 (Talos II platform)
- POWER9 support in coreboot: https://www.youtube.com/watch?v=Mb_SNfMVfw
- Status of OpenPOWER support in coreboot: <https://fosdem.org/2021/schedule/event/statusopenpowercoreboot/>
- Latest releases: https://github.com/3mdeb/openpower-coreboot-docs/blob/release_doc/releases.md
- Development status:
 - documentation: <https://github.com/3mdeb/openpower-coreboot-docs>
 - code: https://github.com/3mdeb/coreboot/tree/talos_2_support

I asked on the qemu_ppc64 state last year (Oct 2020)

- State of qemu_ppc64: <https://lists.yoctoproject.org/g/yocto/message/51158>
 - old layer (2015): <https://github.com/akuster/meta-qemu-bsps>
 - old thread (2014) in linux-yocto:
<https://www.yoctoproject.org/pipermail/linux-yocto/2014-September/003184.html>
 - some unmerged oe-core patches (2014-2016):
<https://patchwork.openembedded.org/project/oe-core/patches/?submitter=11275&state=&q=ppc64&archive=both&delegate=>
- Interesting thread from the Yocto Advisory Board ML (2018) - Future of powerpc support in Yocto Project:
<https://www.yoctoproject.org/pipermail/yocto-ab/2018-May/002226.html>
 - consideration about dropping powerpc (32-bit) support
 - possibly (?) replacing it with ppc64

- Khem Raj pointed me to his patches
 - qemuppc64 machine definition:
<https://github.com/YoeDistro/openembedded-core/commit/885104134403da36b9ecb47ced6423e183262392>
 - powerpc64le support in linux-yocto:
<https://github.com/YoeDistro/openembedded-core/commit/aa9797636a6039ede752a57b05f839ce641e3cfc>
- I tried that and left some status here: <https://github.com/3mdeb/meta-ppc64#notes>
- I stopeed working on that for a while
- Got back a few months later

- Today available in master or hardknott branches
- qemuppc64 was first added to oe-core in March 2021
 - It is based on the POWER9 CPU target

```
commit ef910d7a51a7ef81b725e3cc7b80c2aeb7347ead
Author: Khem Raj <raj.khem@gmail.com>
Date: Tue Mar 9 09:32:06 2021 -0800

    qemuppc64: Add a QEMU machine definition for ppc64

    (From OE-Core rev: 68275b25f0a1941cd9b3d2ddca60e9149ba18d37)

    Signed-off-by: Khem Raj <raj.khem@gmail.com>
    Signed-off-by: Richard Purdie <richard.purdie@linuxfoundation.org>

diff --git a/meta/conf/machine/qemuppc64.conf b/meta/conf/machine/qemuppc64.conf
new file mode 100644
index 000000000000..0682e752be52
--- /dev/null
+++ b/meta/conf/machine/qemuppc64.conf
@@ -0,0 +1,24 @@
+##@TYPE: Machine
+##@NAME: QEMU PPC64 machine
+##@DESCRIPTION: Machine configuration for running a PPC system on QEMU
+
+require conf/machine/include/qemu.inc
+require conf/machine/include/tune-power9.inc
```


- The default tune is ppc64p9le (Little Endian)

```

DEFAULTTUNE ?= "ppc64p9le"

require conf/machine/include/powerpc/arch-powerpc64.inc

TUNEVALID[power9] = "Enable IBM Power9 specific processor optimizations"
TUNE_CCARGS .= "${@bb.utils.contains('TUNE_FEATURES', 'power9', ' -mcpu=power9', '', d)}"

AVAILTUNES += "ppcp9 ppc64p9 ppcp9le ppc64p9le"

TUNE_FEATURES_tune-ppcp9 = "m32 fpu-hard power9 altivec bigendian"
BASE_LIB_tune-ppcp9 = "lib"
TUNE_PKGARCH_tune-ppcp9 = "ppcp9"
PACKAGE_EXTRA_ARCHS_tune-ppcp9 = "${PACKAGE_EXTRA_ARCHS_tune-powerpc} ppcp9"

TUNE_FEATURES_tune-ppc64p9 = "m64 fpu-hard power9 altivec bigendian"
BASE_LIB_tune-ppc64p9 = "lib64"
TUNE_PKGARCH_tune-ppc64p9 = "ppc64p9"
PACKAGE_EXTRA_ARCHS_tune-ppc64p9 = "${PACKAGE_EXTRA_ARCHS_tune-powerpc64} ppc64p9"

TUNE_FEATURES_tune-ppcp9le = "m32 fpu-hard power9 altivec"
BASE_LIB_tune-ppcp9le = "lib"
TUNE_PKGARCH_tune-ppcp9le = "ppcp9le"
PACKAGE_EXTRA_ARCHS_tune-ppcp9le = "${PACKAGE_EXTRA_ARCHS_tune-powerpcle} ppcp9le"

TUNE_FEATURES_tune-ppc64p9le = "m64 fpu-hard power9 altivec"
BASE_LIB_tune-ppc64p9le = "lib64"
TUNE_PKGARCH_tune-ppc64p9le = "ppc64p9le"
PACKAGE_EXTRA_ARCHS_tune-ppc64p9le = "${PACKAGE_EXTRA_ARCHS_tune-powerpc64le} ppc64p9le"

# glibc configure options to get power9 specific library
GLIBC_EXTRA_OECONF += "${@bb.utils.contains('TUNE_FEATURES', 'power9', '--with-cpu=power9', '', d)}"

```

- As for now, there is only one target defined in bsp/qemu-ppc64:

```
# SPDX-License-Identifier: MIT
define KMACHINE qemu_ppc64
define KTYPE standard
define KARCH powerpc

include ktypes/standard.scc

branch qemuppc

include cfg/8250.scc
include features/input/input.scc
include features/usb/ohci-hcd.scc
include features/scsi/disk.scc
include features/scsi/cdrom.scc

include cfg/virtio.scc

include qemu-ppc64.scc

# default policy for standard kernels
include features/latencytop/latencytop.scc
```

- In the qemu-ppc64.cfg we can see that Little Endian is used

```
CONFIG_CPU_LITTLE_ENDIAN=y
```

- We needed to test out the Big Endian build as well

- Two definitions;
 - `mti-malta64-le-standard.scc`
 - `mti-malta64-be-standard.scc`
- KMACHINE definition for BE

```
define KMACHINE mti-malta64-be
define KMACHINE qemumips-64
define KMACHINE qemumips64
```

- KMACHINE definition for LE

```
define KMACHINE mti-malta64-le
define KMACHINE qemumips64le
define KMACHINE qemumips64el
```

- Debian uses `mips64el` for Little Endian and `mips64` for Big Endian
- Debian uses `ppc64el` for Little Endian and `ppc64` for Big Endian
- Fedora uses `ppc64le` for Little Endian and `ppc64` for Big Endian
- openSUSE uses `ppc64le` for Little Endian and `ppc64` for Big Endian
- Gentoo switched from `ppc64` to `ppc64le` for Little Endian:
<https://www.gentoo.org/support/news-items/2020-04-04-new-ppc64le-profiles.html>

Conclusions:

- It may make sense to use `ppc64` for Big Endian and `ppc64le` for Little Endian
 - consistency with major distributions
 - consistency with existing `mips64` target

- By analogy, BE target could be introduced as `qemu-ppc64-be-standard.scc`

```
# SPDX-License-Identifier: MIT
define KMACHINE qemu-ppc64
define KTYPE standard
define KARCH powerpc
```

- And LE target as `qemu-ppc64-le-standard.scc`

```
# SPDX-License-Identifier: MIT
define KMACHINE qemu-ppc64le
define KMACHINE qemu-ppc64el
define KTYPE standard
define KARCH powerpc
```

- Patches: <https://github.com/3mdeb/yocto-kernel-cache/commits/qemu-ppc64be>

- Rename qemu_ppc64 to qemu_ppc64le
- Introduce qemu_ppc64 as Big Endian target

```
#@TYPE: Machine
#@NAME: QEMU PPC64 machine (Big Endian)
#@DESCRIPTION: Machine configuration for running a PPC system on QEMU

require conf/machine/qemu_ppc64le.conf

# change the tune from LE to BE
DEFAULTTUNE = "ppc64p9"
```

- What is the impact of changing qemu_ppc64 to qemu_ppc64le?
 - alternatively, we can just introduce qemu_ppc64be
- Patches: https://github.com/3mdeb/poky/commits/qemu_ppc64be

- Configuration can be found in meta-ppc64 repo:
<https://github.com/3mdeb/meta-ppc64/tree/talos-ii>

- Build

```
$ SHELL=/bin/bash kas-docker build ./meta-ppc64/kas-le.yml
```

- Run QEMU

```
$ SHELL=/bin/bash kas-docker build ./meta-ppc64/kas-le.yml  
(docker)$ runqemu slirp nographic
```

- Result

```
Poky (Yocto Project Reference Distro) 3.3+snapshot-00cd78d9decebcf5201bbdebaef765ecba22f7d0 qemu_ppc64le hvc0  
root@qemu_ppc64le:~# uname -a  
Linux qemu_ppc64le 5.10.34-yocto-standard #1 SMP PREEMPT Mon May 3 02:17:32 UTC 2021 ppc64le ppc64le ppc64le GNU/Linux  
root@qemu_ppc64le:~# echo -n I | od -to2 | head -n1 | cut -f2 -d" " | cut -c6  
1
```

- prints 1 on Little Endian OS

- Configuration can be found in meta-ppc64 repo:
<https://github.com/3mdeb/meta-ppc64/tree/talos-ii>

- Build

```
SHELL=/bin/bash kas-docker build ./meta-ppc64/kas-be.yml
```

- Run QEMU

```
$ SHELL=/bin/bash kas-docker shell ./meta-ppc64/kas-be.yml  
(docker)$ runqemu slirp nographic
```

- Result

```
Poky (Yocto Project Reference Distro) 3.3+snapshot-00cd78d9decebcf5201bbdebaef765ecba22f7d0 qemu_ppc64le hv0  
root@qemu_ppc64le:~# uname -a  
Linux qemu_ppc64le 5.10.34-yocto-standard #1 SMP PREEMPT Mon May 3 02:17:32 UTC 2021 ppc64le ppc64le ppc64le GNU/Linux  
root@qemu_ppc64le:~# echo -n I | od -to2 | head -n1 | cut -f2 -d" " | cut -c6  
0
```

- prints 0 on Big Endian OS

- Available in oe-core in meta/conf/machine/include
 - tune-ppce5500.inc
 - tune-ppce6500.inc
 - tune-power5.inc
 - tune-power6.inc
 - tune-power7.inc
 - tune-power9.inc
- We have POWER5-7 and POWER9 from IBM
- We have e5500 and e6500 from NXP
 - 2011-2012 CPUs
 - NXP shifted to ARM
 - no new PowerPC design since 2013
 - no roadmap for future products

- t1023rdb-64b.conf
 - Board: Freescale T1023RDB
 - SoC: NXP QorIQ T1023
 - CPU: PowerPC e5500 from NXP
- t4240rdb-64b.conf
 - Board: Freescale 4240RDB
 - SoC: NXP QorIQ T4240
 - CPU PowerPC e6500 from NXP
- They are using ppce5500 and ppce6500 tune by default
 - Big Endian
 - no Little Endian variant

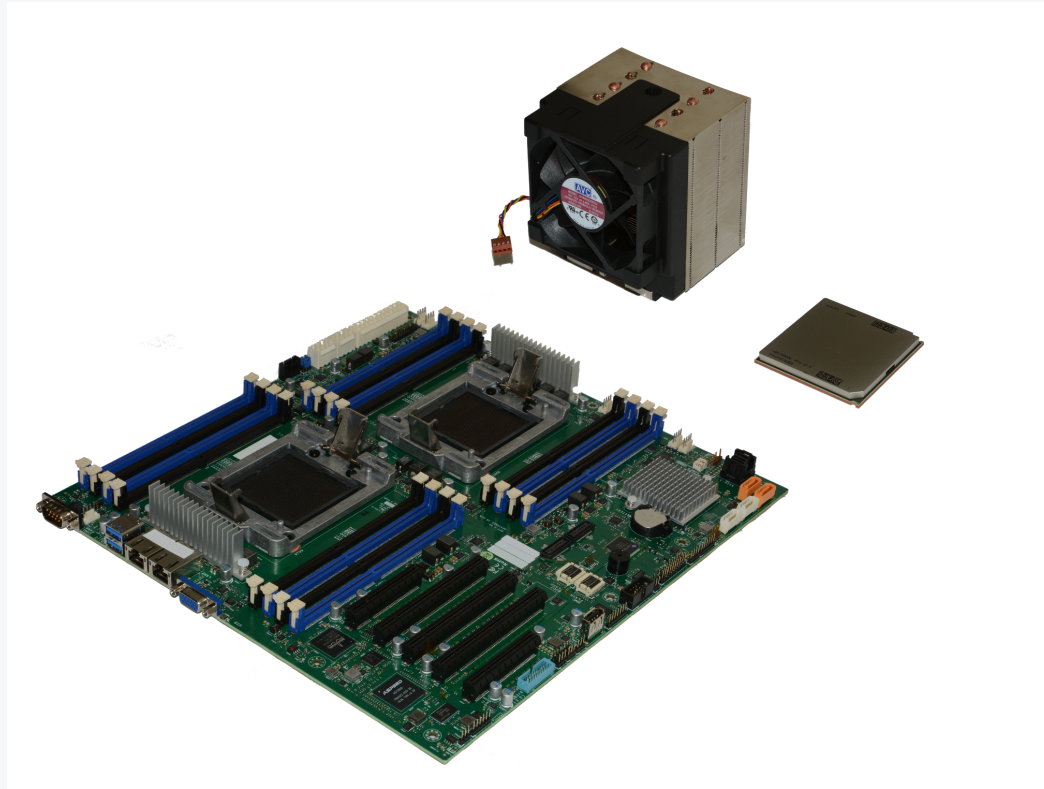
Advertised as:

- The first modern (post-2013), owner-controllable mainboard
- Built around the IBM POWER9 processor
- Open source firmware, all the way down to the CPU microcode

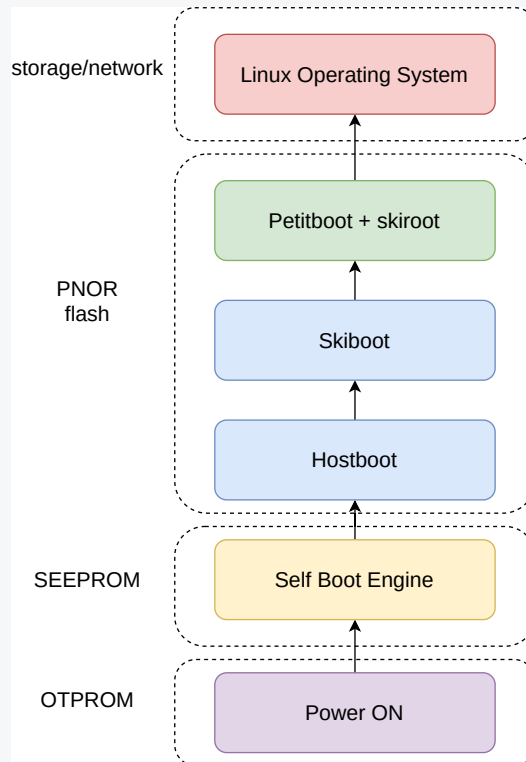
Main specs:

- 2 x POWER9-compatible CPU sockets
- 16 x DDR4 DIMM slots (up to 2TB DDR4 total memory)
- 3 x PCIe 4.0 x16 slots
- 2 x PCIe 4.0 x8 slots
- 2 x Broadcom Gigabit Ethernet ports
- 4 x USB 3.0 ports, 1 x USB 2.0 port
- 1 x ASpeed (AST2500) BMC with OpenBMC:

<https://github.com/openbmc/openbmc>



<https://static.raptorcs.com/TL2B01/images/boardlarge.png>



- skiroot
 - Linux kernel and rootfs
 - provides environment where Petitboot runs
 - built with Buildroot
- Petitboot - <https://github.com/open-power/petitboot>
 - kexec-based bootloader
 - Linux kernel uses kexec syscall to load another kernel
 - pb-discover - manages devices, finds and boots boot targets
 - petitboot-nc - ncurses client (UI)
 - parses syslinux, kboot, yaboot, pxe and native (petitboot) configs into a single menu

- Based on the qemu64 target

```
#@TYPE: Machine
#@NAME: Talos II machine
#@DESCRIPTION: Machine configuration for running on Talos II (POWER9) hardware

require conf/machine/include/tune-power9.inc

PREFERRED_PROVIDER_virtual/kernel ?= "linux-mainline"

KERNEL_IMAGETYPE = "vmlinux"

SERIAL_CONSOLES ?= "115200;hvc0"

MACHINE_FEATURES = "alsa usb gadget screen keyboard pci usbhost ext2 ext3 serial vfat"

MACHINE_EXTRA_RRECOMMENDS += " kernel-modules"

IMAGE_FSTYPES += "tar.bz2 ext4"

PREFERRED_PROVIDER_virtual/xserver ?= "xserver-xorg"
PREFERRED_PROVIDER_virtual/egl ?= "mesa"
PREFERRED_PROVIDER_virtual/libgl ?= "mesa"
PREFERRED_PROVIDER_virtual/libgles1 ?= "mesa"
PREFERRED_PROVIDER_virtual/libgles2 ?= "mesa"

XSERVER ?= "xserver-xorg \
    ${@bb.utils.contains('DISTRO_FEATURES', 'opengl', 'mesa-driver-swrast xserver-xorg-extension-glx', '', d)} \
    xf86-video-fbdev \
    "

IMAGE_INSTALL_append = " petitboot"
```

- petitboot.bb

```
SRC_URI = "file://petitboot.conf"
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COREBASE}/meta/COPYING.MIT;md5=3da9cfbcb788c80a0384361b4de20420"
S = "${WORKDIR}"

do_install() {
    install -d -m 0644 ${D}/boot/petitboot.conf
    install -m 0644 ${S}/petitboot.conf ${D}/boot/petitboot.conf
}

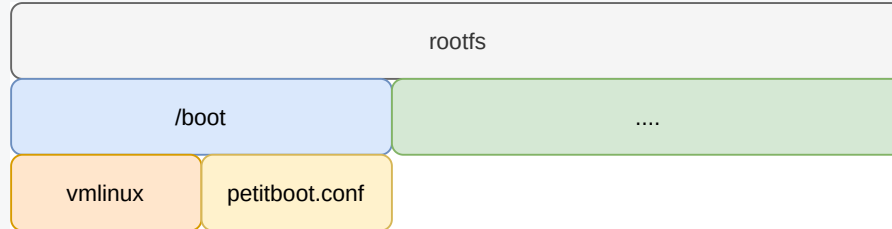
FILES_${PN} += "/boot"
```

- petitboot.conf

```
name 3mdeb Yocto build for PPC64
image /boot/vmlinux
args console=tty0 console=hvc0 rootdelay=3 rootwait panic=10 root=/dev/sdb
```


- linux-mainline 5.12.0 from:
 - <https://gitlab.com/pbarker.dev/meta-linux-mainline/-/tree/4b288396eff43fe9b1a233aed1ce9b48329a2eb6>
- in-tree defconfig

```
KBUILD_DEFCONFIG = "powernv_defconfig"  
KCONFIG_MODE = "alldefconfig"
```



- A few attempts have been made to boot up partitioned image via Petitboot
 - created manually
 - created via wic tool
- All of them failed
 - failed to mount the device by skirroot
- Success with manual ISO creation so far only
 - some further investigation is needed why the wic image does not work


```
$ tar xf core-image-full-cmdline-talos-ii-le.tar.bz2 -C rootfs
$ mkisofs -R -o power9-image.iso rootfs
```

[Return to OpenBmc](#)

```
1.00 A1000640
[USB: sdb / 2021-05-11-22-15-35-00]
* 3mdeb Yocto build for PPC64
```

```
System information
System configuration
System status log
Language
Rescan devices
Retrieve config from URL
Plugins (0)
Exit to shell
```

```
Loaded kernel image from file:///var/petitboot/mnt/dev/sdb/boot/
Performing kexec load
```

 Return to OpenBmc

```
g /var/lib/nfs/v4recovery as the NFSv4 state recovery directory
[ 37.175224][ T474] NFSD: Using legacy client tracking operati
ons.
[ 37.175233][ T474] NFSD: starting 90-second grace period (net
f000001f)
done
starting mountd: done
Starting system log daemon...0
Starting crond: OK

Poky (Yocto Project Reference Distro) 3.3+snapshot-5113b3d5e7ec73
dfb1ca561424c4c574174d0dd4 talos-ii-le hvc0





talos-ii-le login: root
root@talos-ii-le:~# uname -a
Linux talos-ii-le 5.12.0 #1 SMP Sun Apr 25 20:49:08 UTC 2021 ppc6
4le ppc64le ppc64le GNU/Linux
root@talos-ii-le:~# cat /etc/issue
Poky (Yocto Project Reference Distro) 3.3+snapshot-5113b3d5e7ec73
dfb1ca561424c4c574174d0dd4 \n \l

root@talos-ii-le:~#
```

- Add machine configuration - **DONE**
- Add petitboot configuration recipe - **DONE**
 - Petitboot supports grub or syslinux configs
 - maybe we should reuse that configuration instead
- Select kernel provider and config - **DONE**
 - linux-mainline and pownv_defconfig
- Automate creation of compatible image layout - **IN PROGRESS**
 - only ISO format worked so far
- Create a proper BSP layer - **TODO**
 - should it be a separate layer or upstreamed somewhere?
- Talos II patches: <https://github.com/3mdeb/meta-ppc64/commits/talos-ii>

- Defeating Invisible Enemies:Firmware Based Security in OpenPOWER Systems:
 - <http://events17.linuxfoundation.org/sites/events/files/slides/op-stboot-lss-2017-v7.0.pdf>
- Petitboot - Four Years of Linux as a Bootloader:
 - <https://2018.osfc.io/uploads/talk/paper/9/petitboot.pdf>
- Tell Me About Petitboot:
 - <https://sthbrx.github.io/blog/2016/05/13/tell-me-about-petitboot/>

We are open to cooperate and discuss

-  contact@3mdeb.com
-  facebook.com/3mdeb
-  [@3mdeb_com](https://twitter.com/_@3mdeb_com)
-  linkedin.com/company/3mdeb
- <https://3mdeb.com>
- [Book a call](#)
- [Sign up for the newsletter](#)

Feel free to contact us if you believe we can help you in any way. We are always open to cooperate and discuss.

Q&A